- **GNU / Linux** es un SO tipo Unix libre. Fue, y es, diseñado por programadores de todo el mundo.
    - **GNU** → Versión de UNIX libre.
        - Cualquier propósito.
        - Estudiar su funcionamiento.
        - Distribuir sus copias.
        - Mejorar programas.
- Es libre y gratuito.
- Existe una versión base y diversas distribuciones.
    - Incluye una versión de Kernel y programas con configuraciones, modificando la versión base.
- Código abierto, posibilitando su visualización, análisis, modificación, etc.
- **UNIX** → familia de SO pagos. GNU/Linux se desprende.
- **Multiusuario** → puede ser utilizado simultáneamente por múltiples usuarios, con sus permisos y configuraciones.
- **Multitarea** → se pueden ejecutar y administrar varios procesos o aplicaciones simultáneamente.
- **Multiprocesador** → utiliza múltiples procesos para realizar tareas de procesamiento simultáneamente. Distribuye la carga de trabajo mejorando el rendimiento y capacidad.
- **Altamente portable** → fácilmente trasladado y ejecutado en diferentes entornos y HW.
- Diversos intérpretes de comandos.
- Manejo de usuarios y permisos.
- Los directorios pueden estar en particiones diferentes.
- **POSIX** → estándar que define interfaces y comportamientos de SO Unix. Garantiza portabilidad de programas y scripts entre SO.
- **ESTRUCTURA** → **Núcleo**, **Intérprete de Comandos**, y **Sistema de Archivos.**
    - **Núcleo** → Kernel. Ejecuta programas y gestiona dispositivos HW. Posibilita el trabajo entre SW y HW, administrando memoria, CPU, y E/S.
        - Monolítico híbrido.
            - ***Monolítico*** → incluye todos los controladores y servicios en un solo núcleo de SO, administrando HW, memoria, procesos, y su comunicación.
            - ***Híbrido*** → modularidad y flexibilidad de un micronúcleo, y la estructura de un núcleo monolítico, buscando un balance entre rendimiento y capacidad de extensión.
        - Es posible tener múltiples en la misma máquina.
        - Se encuentra en /boot.

    - **Intérprete de Comandos** → Shell. Comunicación entre usuario y SO. Ejecuta programas a partir de comandos. Cada usuario tiene su Shell, programable y personalizable.
    - **Sistema de Archivos** → FileSystem. Organizan la forma de almacenamiento de archivos.
        - El directorio **/** es la raíz del sistema.
        - **/home** → archivos de usuarios, uno por cada uno.
        - **/var** → información que varía de tamaño.
        - **/etc** → archivos de configuración.
            - Definen el comportamiento de los servicios y el sistema.
        - **/bin** → binarios y ejecutables.
            - Operaciones básicas.
            - Accesible para todos los usuarios.
        - **/dev** → enlaces a dispositivos.
            - Permite al SO interactuar con el HW como si fuera un archivo.
        - **/usr** → aplicaciones de usuarios.
        - ***FHS*** → File Hierarchy System. El estándar de jerarquía en sistemas de archivos.
- **PARTICIONES** → divide lógicamente el disco físico, mejorando el rendimiento.
    - Cada SO se instala en una partición diferente, facilitando su administración.
    - Cada partición se formatea con un tipo de filesystem destino.
    - Separar los datos de usuario de las aplicaciones y SO.
    - Kernel en partición de sólo lectura o que no se monta (no disponible para usuarios).
    - Facilita el mantenimiento y recuperación de datos; y el formateo y reinstalación de SO sin afectar archivos personales.
    - Diferentes permisos por partición.
    - Permite asignar un tamaño específico a ciertas áreas, evitando que un tipo de datos llene el disco.
        - Esto puede provocar que una partición esté llena y otra vacía, sin poder utilizarla.
        - Mayor complejidad de gestión.
        - Riesgo de pérdida de datos si se modifican particiones incorrectamente.
        - Incompatibilidad de FileSystems en diferentes SO.
        - Pueden ralentizar el sistema si se usan inadecuadamente.
    - ***Partición primaria*** → división cruda del disco.
        - Hasta 4.
        - Se almacena información de la misma en el MBR.
    - ***Partición extendida*** → contiene unidades lógicas.

        - Solo 1.
        - No se define un tipo de FS.
    - ***Partición lógica*** → ocupa la totalidad o parte de la extendida y se le define un tipo de FS.
        - Se conectan como lista enlazada.
    - Para crearlas se usa un SW llamado **particionador**, que puede ser:
        - ***Destructivo*** → crea y elimina particiones (fdisk).
        - ***No destructivo*** → crea, elimina, y modifica particiones (fips, gparted).
    - Se identifican según el tipo de controlador de disco y el número de partición.
    - Se representan como archivos de dispositivos en /dev.
        - **IDE** → son más antiguos y se identifican con el prefijo hd.
            - **/dev/hda** : 1° disco duro IDE.
            - **/dev/hdb** : 2° disco duro IDE.
                - **/dev/hda1** : 1° partición del 1° disco IDE.
                - **/dev/hda2** : 2° partición del 1° disco IDE.
                - **/dev/hdb2** : 2° partición del 2° disco IDE.
            - Generalmente las primeras 4 particiones son primarias o extendidas, y el resto lógicas.
        - **SCSI y SATA** → modernas. Utilizan el prefijo sd.
            - **/dev/sda** : 1° partición disco SCSI o SATA.
            - **/dev/sdb** : 2° partición disco SCSI o SATA.
            - **/dev/sda1**
            - **/dev/sda2**
    - Una partición es obligatoria, pero se recomiendan al menos dos.
        - **Obligatoria - Partición Raíz:**
            - ***Tipo***: Primaria
            - ***Identificación***: 83 (MBR), u 8300 (GPT).
            - ***Sistemas de archivos***: EXT4 (recomendado), pero se pueden usar otros.
            - ***Punto de Montaje***: **/** (raíz del FS).
        - **Recomendada - Raíz + SWAP:**
            - Raíz ya mencionada.
            - ***Tipo***: Primaria o Lógica.
            - ***Identificación***: 82 (MBR), u 8200 (GPT).
            - ***Sistema de archivos***: no aplica ya que es área de intercambio.
            - ***Punto de Montaje***: no tiene, se utiliza por el sistema como espacio de intercambio.

        - **SWAP** se utiliza para aumentar la memoria virtual del sistema, permitiendo la utilización del disco como extensión de la RAM.
            - <4GB RAM, SWAP ≤ RAM.
            - 4GB RAM, SWAP ≥ RAM.
    - **Opcional pero Recomendada - Raíz + SWAP + Home:**
        - Raíz y SWAP ya mencionadas.
        - ***Tipo***: Primaria o Lógica.
        - ***Identificación***: 83 (MBR) u 8300 (GPT).
        - ***Punto de Montaje***: **/home**.

- **ARRANQUE / BOOTSTRAP:**
    - BIOS (Basic I/O System) → responsable de iniciar la carga del SO a través del MBC.
        - Proceso de arranque.
        - Configuración del sistema.
        - Interfaz entre HW y SO.
        - Compatibilidad.
    - UEFI (Unified Extensible Firmware Interface) → interfaz de firmware moderna que reemplaza el BIOS.
        - Arranque del sistema.
        - Configuración del hardware.
        - Compatibilidad y flexibilidad.
    - MBR → estructura de datos y código en el primer sector (0) de un disco duro.
        - Código de arranque.
        - Tabla de particiones.
        - Identificación del disco.
    - MBC → código en el MBR responsable del proceso de arranque.
        - Se ejecuta al encender la computadora.
        - Carga el gestor de arranque desde una partición activa, y pasa el control a este para cargar el SO.
    - GPT (Guid Partition Table) → esquema de particionamiento que sustituye el MBR.
        - Formato:
            - GUID → Identificadores Únicos Globales por partición.
            - Tabla de Particiones Principal y de Respaldo.
            - Capacidad de particiones casi ilimitado.
            - Soporta discos de hasta 9.4 zebibytes.

- Gestor de Arranque (bootloader) → carga una imagen del Kernel (SO) de alguna partición para su ejecución.
  - Se ejecuta luego del BIOS.
  - Dos modos de instalación:
    - En MBR.
    - En el sector de arranque de la partición raíz o activa.
- Proceso de bootstrap → Pasos desde encendido computadora hasta que se carga el SO.
  - Encendido.
  - POST: Verificación del HW por el BIOS / UEFI.
  - Inicialización del HW: configuración de los componentes esenciales.
  - Configuración del Orden de Arranque: transferencia del control desde el dispositivo de arranque al gestor de arranque.
  - Ejecución del Gestor de Arranque: carga del núcleo del SO.
  - Inicialización de Servicios del Sistema.
  - Inicio de la Interfaz de Usuario.
- Proceso de shutdown:
  - Recepción de la orden de apagado.
  - Notificación a los usuarios.
  - Detención de servicios.
  - Desmontaje del sistema de archivos (guardado).
  - Apagado del HW.
  - Apagado completo.

- **ARCHIVOS:**
  - Se identifican a través de:
    - Ruta del archivo (Ejemplo → /home/usuario/documentos/archivo.txt)
    - Nombre → “archivo”
    - Número de Inode → estructura que almacena metadatos del archivo como tamaño, fecha de creación, etc.
    - Permisos y Propietario.
  - Editores de Texto:
    - vi → el más popular. Tiene dos modos principales.
      - Modo de Comando: ejecutar comandos para manipular el texto.
      - Modo de Inserción: editar el texto.
    - mcedit → basado en el entorno Midnight Commander, con una interfaz de usuario más amigable.

        - Interfaz de usuario basada en texto, pero más intuitiva, con menús y comandos visibles.
    - Comandos como cat, more, o less, solo permiten visualizar archivos, mientras que con editores de texto se puede crear y modificar a los mismos.

- **PROCESO DE ARRANQUE SYSTEMV:**
    - Pasos:
        - Se empieza a ejecutar el código del BIOS.
        - El BIOS ejecuta el POST.
        - El BIOS lee el sector de arranque (MBR).
        - Se carga el gestor de arranque (MBC).
        - El bootloader carga el kernel y el initrd.
        - Se monta el initrd como sistema de archivos raíz y se inicializan componentes esenciales.
        - El kernel ejecuta el proceso init y se desmonta el initrd.
        - Se lee el /etc/inittab.
        - Se ejecutan los scripts apuntados por el runlevel 1.
        - El final del runlevel 1 le indica que vaya al runlevel por defecto.
        - Se ejecutan los scripts apuntados por este.
        - El sistema está listo para usarse.
    - Proceso INIT → carga todos los subprocesos necesarios para el funcionamiento del SO. Monta los FS y hace disponibles los demás dispositivos.
        - Lo ejecuta el Kernel.
        - Posee PID 1 y se encuentra en /etc/inittab.
        - No tiene padre, y es padre de todos los procesos.
    - Runlevels → modo en el que arranca Linux. Divisiones del proceso de arranque.
        - Cada uno es responsable de levantar o bajar servicios.
        - Existen 7, permiten iniciar procesos de arranque o apagado del sistema.
        - 0: halt.
        - 1: single user mode.
        - 2: multiuser without NFS (soporte de red).
        - 3: full multiuser mode console.
        - 4: -
        - 5: X11. Modo multiusuario completo con login gráfico basado en X.
        - 6: reboot.

        - El runlevel a ejecutar en el inicio del SO se define en el archivo /etc/inittab → id:5:initdefault
    - Archivo /etc/inittab → configura el proceso de inicialización y gestión de niveles de ejecución en sistemas SysV init.
        - Se almacenan niveles de ejecución, procesos y servicios a iniciar, y configuración de consolas.
        - Estructura → identificador, niveles de ejecución, acción, proceso o comando
            - id:nivelesEjecucion:acción:proceso
    - Scripts RC → gestionan el inicio y detención de servicios en sistemas SysV init.
        - Ejecutan comandos para preparar el sistema para el funcionamiento completo en el nivel de ejecución deseado.
        - Inician servicios y configuraciones durante el arranque, y detienen servicios y realizan tareas de limpieza durante el apagado o reinicio.
        - Se almacenan en /etc/init.d
            - Si corresponden a un nivel de ejecución → /etc/rc.d o /etc.rc*.d
        - Se ejecuta en función de un prefijo S (start) o K (kill) seguido de un número que indica orden de ejecución.
        - Cuando el sistema se detiene o cambia nivel de ejecución, se ejecutan los scripts para detener los servicios en orden inverso.
- SystemD → sistema que centraliza la administración de demonios y librerías del sistema.
    - Mejora el paralelismo de booteo.
    - Puede ser controlado por systemctl.
    - Compatible con SysV.
    - Reemplaza al proceso init.
    - Unit → unidades de trabajo.
        - Service → controla un servicio.
        - Socket → encapsula IPC, socket de sistema, o FS FIFO.
        - Target → agrupa units o establece puntos de sincronización durante el booteo,
        - Snapshot → almacena el estado de un conjunto de unidades para ser restablecido más tarde.
    - Target → reemplazan los runlevels.

- **USUARIOS:**
    - Archivos para guardar su información:

        - /etc/passwd → info básica.
        - /etc/shadow → contraseñas cifradas y detalles de autentificación.
        - /etc/group → info sobre grupos.
        - /etc/gshadow → contraseñas de grupos y administración.
    - UID → User Identifier.
        - Identificador único asignado a cada usuario.
    - GID → Group Identifier.
        - Identificador único asignado a cada grupo.
    - Usuario root → administrador del sistema. Posee permisos para realizar toda tarea sobre el sistema y acceder a todo archivo.
        - Su UID es 0.

- **FILESYSTEM:**
    - Los permisos para definir quién puede leer, escribir, o ejecutar un archivo o directorio se dividen en:
        - Lectura (read)
        - Escritura (write)
        - Ejecución (execute)
    - Entidades:
        - Usuario (owner)
        - Grupo (group)
        - Otros (others)
    - Comandos de Permisos.
    - Full Path Name → nombre de ruta completo. Ubicación absoluta.
        - /home/usuario/documentos/archivo.txt
    - Relative Path Name → nombre de ruta relativo. La ruta basada desde la ubicación actual.
        - Estado en /home/usuario → documentos/reporte.txt
    - Comandos FS.

- **PROCESOS** → programa en ejecución por el SO.
    - Incluye su código, estado, variables, recursos, y entorno de ejecución.
    - PID → Process ID.
        - Identificador único asignado a un proceso en ejecución.
    - PPID → Parent Process ID.
        - Identificador del proceso padre.
    - Atributos:
        - PID.

        - PPID.
        - UID.
        - EUID → Effective User ID. Determina privilegios.
        - GID.
        - Prioridad.
        - Uso de Memoria.
        - Tiempo de CPU.
        - Archivo ejecutable.
        - Entorno.
        - Recursos abiertos.
    - Comandos procesos.
    - Foreground → primer plano. Vinculado directamente a la terminal desde la cual fue lanzado.
        - Ocupa la terminal, bloqueando su uso.
        - El usuario puede interactuar directamente con el proceso.
        - Es posible llevarlo a background.
            - Ctrl + Z y bg.
    - Background → segundo plano. Ejecutando de manera independiente a la terminal.
        - La terminal queda libre para su uso.
        - No interactúa directamente con la terminal, no requiere entrada del usuario, y su salida no se muestra en la terminal.
        - Es posible llevarlo al foreground.
            - fg o fg %n
        - & al final del comando.
    - Pipe (|) → permite la comunicación entre procesos desde la shell.
        - Conecta stdout (salida estándar) del primer comando con la stdin (entrada estándar) del segundo.
            - $ ls | more → la salida de ls es enviada como entrada de more.
    - Redirecciones → gestionan la entrada / salida de comandos, guarda resultados, procesa datos entre comandos, y maneja errores por separado.
        - Destructivas → >
            - Si el archivo destino no existe, lo crea.
            - Si existe, se lo trunca y se escribe el nuevo contenido.
        - No destructivas → >>
            - Si el archivo destino no existe, lo crea.
            - Si existe, se agrega la información al final.
        - De salida estándar (stdout).
        - De entrada estándar (stdin).
        - De errores estándar (stderr).

        - Combinada.
        - Tuberías / Pipes.
    - Más comandos.
- OTROS:
- Empaquetar archivos → agrupar múltiples archivos y directorios en un solo archivo, sin necesariamente comprimir su contenido.
    - Facilita su manejo, siendo más sencillo transferir, almacenar, o respaldar un único archivo.
    - Es posible empaquetar y comprimir.
    - Comandos de empaquetamiento y compresión.
- Shell Scripting → escribir scripts o programas usando la Shell de un SO.
    - Automatización de tareas.
    - Aplicaciones interactivas.
    - Aplicaciones con interfaz gráfica.
    - Comandos!! (variables también)